2015 IEEE 27th International Conference on Tools with Artificial Intelligence

An Equitable Solution to the Stable Marriage Problem

Ioannis Giannakopoulos*, Panagiotis Karras†, Dimitios Tsoumakos‡ §, Katerina Doka* and Nectarios Koziris* §
* Department of Electrical and Computer Engineering, National Technical University of Athens, Greece
{ggian, katerina, nkoziris}@cslab.ece.ntua.gr
† Skolkovo Institute of Science and Technology, Russia
karras@skoltech.ru
‡ Department of Informatics, Ionian University, Greece
dtsouma@ionio.gr
§ Athena Research and Innovation Center in ICKT, Greece
{dtsouma,nkoziris}@imis.athena-innovation.gr

***Abstract*—A stable marriage problem (SMP) of size n involves n men and n women, each of whom has ordered members of the opposite gender by descending preferability. A solution is a perfect matching among men and women, such that there exists no pair who prefer each other to their current spouses. The problem was formulated in 1962 by Gale and Shapley, who showed that any instance can be solved in polynomial time, and has attracted interest due to its application to any two-sided market. Still, the solution obtained by the Gale-Shapley algorithm is favorable to one side. Gusfield and Irving introduced the *equitable stable marriage problem* (ESMP), which calls for finding a stable matching that *minimizes* the distance between men's and women's sum-of-rankings of their spouses. Unfortunately, ESMP is strongly NP-hard; approximation algorithms therefor are impractical, while even proposed heuristics may run for an unpredictable number of iterations. We propose a novel, *deterministic* approach that treats both genders equally, while eschewing an exhaustive exploration of the space of all stable matchings. Our thorough experimental study shows that, in contrast to previous proposals, our method not only achieves high-quality solutions, but also terminates efficiently and repeatably on all tested large problem instances.**

I. Introduction

The *stable marriage problem* (SMP) [1], [2] pertains to matching the agents in two sets of size n, where each agent maintains a preference ranking over those in the other set. Variants of the problem appear in any two-sided market, as in matching doctors to hospitals [3], [4], students to schools [5], or sailors to ships [6]. In the classic formulation, the two sets consist of men and women in a marriage market, while the problem calls for matching men and women to each other in such a way that there exist no two people who would both rather be married to each other than to their assigned spouses. When no such pair exists, the whole matching is said to be *stable*. The problem was formulated by Gale and Shapley [7], who proved that a stable solution can always be arrived at by an $O(n^2)$ algorithm.

Unfortunately, the Gale-Shapley Stable Marriage Algorithm (SMA) only yields highly satisfied men and dissatisfied women, or vice versa, namely a male optimal or female optimal solution [8]; therefore it is inappropriate for most real-world applications, which require matchings to be not only stable, but also equitable [1], [9]. Thus, the *equitable stable marriage problem* (ESMP) has attracted attention on its own [10]–[15]. The ESMP calls for finding a stable matching that satisfies a notion of equity between the two sides of the examined market. Such equity implies avoiding unequal degrees of *happiness* among the two sides. At the same time, equity and overall happiness are not necessarily compatible, thus, equity should not be achieved at the cost of sacrificing the overall happiness. A solution that achieves equity by yielding equally miserable men and women would not be attractive. Unfortunately, the problem of optimizing equity alone is NP-hard [10]. Recent research has provided approximation algorithms [11] or heuristics [12]–[15] for this problem. Still, to date there is no practical algorithm that efficiently yields equitable stable matchings for large problem instances.

In this paper, we provide an efficient and practical solution for the ESMP. We eschew an exhaustive search over stable matchings, and instead build on the Gale-Shapley algorithm in a manner that treats both sides equitably and still achieves stable solutions. Achieving such equity is a nontrivial problem. We conduct a thorough experimental study of our techniques on much larger data than those of previous studies, compared against a recent alternative [15] that does *not* always terminate, as well as to the Gale-Shapley algorithm.

II. Background and Related Work

An instance I of the *stable marriage problem* (SMP) consists of n men and n women, where each person has a *preference list* that strictly orders all members of the opposite gender. If a man m prefers w_1 to w_2, we write $w_1 \succ_m w_2$; a similar notation is applied to women's preferences. A *perfect matching* M on I is a set of disjoint man-woman pairs on I. When a man m and a woman w are matched to each other in M, we write $M(m) = w$ and $M(w) = m$. A man m and a woman w are said to form a *blocking pair* for M (or to *block* M) when: (i) $M(m) \neq w$; (ii) $w \succ_m M(m)$; and (iii)

1082-3409/15 $31.00 © 2015 IEEE
DOI 10.1109/ICTAI.2015.142

$m \succ_w M(w)$. A matching M is *unstable* if a blocking pair exists for M, and *stable* otherwise. The SMP calls for finding a stable matching M.

A. The Gale-Shapley Algorithm

The standard algorithm for solving the SMP, proposed by Gale and Shapley [7], goes through a series of iterations. At iteration i, each single man proposes to his most highly preferred woman to whom he has not yet proposed, who may or may not be already engaged; subsequently, each woman considers all proposals she received from suitors, as well as her current fiancé, if such exists, and accepts (or retains) an engagement to the most preferable among them. This process goes on until all men (and, in consequence, all women) become engaged. At that point all marriages are final and stable [1].

The Gale-Shapley algorithm terminates successfully after a quadratic number of steps (yet linear in the input size of preference lists). During its operation, once a woman becomes engaged she remains engaged, while she may improve her position by rejecting one fiancé for another. On the other hand, an engaged man may be abandoned by his spouse and become single again. A woman's status changes only once from single to engaged, and thereafter her ranking of her spouse can only improve, while a man's status may change back and forth multiple times, while the ranking of his assigned spouse can only worsen. A bachelor m never has to look back to a woman w who has already rejected him; once w rejects m, she will never prefer him to than any future spouse. Thus, the algorithm is characterized by a monotonicity of men's and women's rankings of their assigned fiancés. Nevertheless, even though women can reject one fiancé for another, women passively react to men's proposals, while men actively make proposals on their own. In effect, the algorithm caters to the well-being of men and not to that of women; in the matching it generates, each man gets the highest, and each woman the lowest, preference they could get in any stable solution; this matching is male-optimal and female-pessimal [8].

B. Quality Metrics and Previous Solutions

Even while the matching returned by the Gale-Shapley algorithm is an extreme proposer-optimal solution, any instance I of the problem may admit many different stable matchings. One can reasonably try to achieve a matching that is not only stable, but also judged to be good by some quality metric. Past research has defined three quality criteria. Let $pr_m(w)$ (respectively, $pr_w(m)$) denote the position of woman w in man m's preference list (respectively, of m in w's list). The *regret cost* $r(M)$ of a stable matching M is:

$$r(M) = \max_{(m,w)\in M} \max\{pr_m(w), pr_w(m)\} \qquad (1)$$

On the other hand, the *egalitarian cost* $c(M)$ is:

$$c(M) = \sum_{(m,w)\in M} pr_m(w) + \sum_{(m,w)\in M} pr_w(m) \qquad (2)$$

Finally, the *sex equality cost* is defined as:

$$d(M) = \left| \sum_{(m,w)\in M} pr_m(w) - \sum_{(m,w)\in M} pr_w(m) \right| \qquad (3)$$

Gusfield and Irving [1] formulated the optimization problems of finding stable matchings that minimize each of these quality metrics. These are tough problems, as admissible stable matchings grow exponentially in the problem instance size [16]. Nevertheless, Gusfield [17] proposed an $O(n^2)$ algorithm for the minimum-regret SMP, while Irving et al. [18] provided a $O(n^4)$ solution for the minimum-egalitarian SMP, later improved to $O(n^3)$ by Feder [19]. These algorithms exploit a lattice represented by a polynomial-size rotation poset [20], containing all stable matchings. In contrast, there is no known polynomial-time solution for minimizing the sex equality cost. Kato [10] proved that this problem is strongly NP-hard, while Iwama et al. [11] provided an $O\left(n^{3+\frac{1}{\epsilon}}\right)$ approximation algorithm that obtains a stable matching M such that $d(M) \leq \epsilon\Delta$, for a given constant ϵ, where $\Delta = \min\{d(M_0), d(M_z)\}$, M_0 the man-optimal and M_z the woman-optimal stable matching; a $O\left(n^{3+2\frac{1+\epsilon}{\delta}}\right)$ variant bounds both $d(M)$ and $c(M)$ in terms of ϵ and δ. Gelain et al. [12], [21] provided a local search algorithm that finds an arbitrary stable marriage other than the male-optimal or female-optimal ones for small problem sizes, yet did not provide any results on quality metrics. Aldershof et al. [22] have shown how to refine the set of linear inequalities that describe the stable matching polytope and proposed a randomized procedure that results in stable matchings that do not favor one group over the other, yet does not attempt to optimize a cost metric. In [23], Roth and Vande identified that starting from an arbitrary matching and allowing randomly chosen blocking pairs to match, the final matching will be stable with probability equal to one. This interesting approach examines the idea of transforming a possibly not stable matching and transforming it into a stable matching. However, there exist no guarantees regarding the equality aspect of the matching, since the choice of blocking pairs must be random and also the execution time of this methodology may be prohibitive for big data sizes. Furthermore, in [24], Ma demonstrated that not all possible stable matchings are reachable from any random starting position.

Most recently, Everaere et al. [14], [15] proposed heuristics that achieve equitable stable marriages, culminating in *Swing* [15], a deterministic algorithm that allows both sides to repetitively issue proposals; yet, unfortunately, Swing invests a $O(n^2)$ time complexity per iteration, and may run for an unpredictable number of such iterations even on small data sizes. Thus, no efficient algorithm that provides a solution of low sex equality cost exists. *Swing*++ [25] tries to solve the non-termination problem of Swing by detecting repetitive patterns of proposals and isolating the responsible agents; unfortunately, it constitutes a step backward rather than a step forward from Swing: it still does not guarantee termination, while it also burdens the algorithm's runtime with elaborate cycle-detection and stability-checking mechanisms. Similarly in [26] the problem is represented in the form of a "marriage table" and the proposed algorithm named ZigZag, targets to compromise global satisfaction, sex equality and stability. This approach guarantees termination, but does not guarantee stability. We strive to provide such algorithms that achieve both equity, i.e., low $d(M)$ and high overall happiness, i.e., low $c(M)$. Other works have studied variants of the SMP in the presence of preference ties and incomplete preference lists [27], [28], agents having incentives to misreport their preferences [29]–[31], partially ordered preferences [32], and weaker notions of stability [33].

III. Equitable Stable Marriage

We now set out to devise an algorithm that provides an equitable solution to the SMP. As discussed, the Gale-Shapley algorithm assigns two strictly defined and mutually exclusive roles to the problem's two sides. One side is designated as *proposers*, assuming an active role, while the other side is relegated to the role of *acceptors*, expecting and reacting to the proposers' initiatives. In consequence, the outcome of the algorithm is proposer-optimal and acceptor-pessimal. In an extreme case, assume that each man m_i ranks woman w_i as his first preference: $pr_{m_i}(w_i) = 1$, while w_i ranks m_i as her last preference: $pr_{w_i}(m_i) = n$, for $i \in [1, n]$. Then, the outcome of the algorithm will be a stable matching M where $M(m_i) = w_i$, $i \in [1, n]$; thus, all women obtain their last choice, while all men obtain their first choice. We reason that, in order to obtain a more equitable solution to the SMP, the algorithm should treat both sides in an equitable and non-discriminatory manner. Ideally, the algorithm should be gender-blind, and the roles of the two sides interchangeable with each other. Thus, we allow both genders (groups) to act as both proposers and acceptors. In particular, at each iteration of the algorithm, we should assign the role of proposer to the one side and that of acceptor to the other. In the following we analyze the mechanics of this approach and the dilemmas it raises, and show how a stable matching can still be obtained in this setting.

A. A Gender-Neutral Proposal

To render our discussion gender-neutral, we present it in terms of two groups of size n, A and B, where $a_i \in A$ and $b_i \in B$ for $i \in [1, n]$. Let ℓ_{a_i} be the preference list of agent a_i and ℓ_{b_i} that of b_i; each preference list holds the desired spouses of each agent ordered by their rank. For instance, $\ell_{b_4}[10] = a_2$ means that a_2 is the 10^{th} preference of b_4. We express the same relationship as $pr_{b_4}[a_2] = 10$. If a_i prefers b_j with rank c, then $\ell_{a_i}[c] = b_j$ and $pr_{a_i}[b_j] = c$. We postulate that the role of proposers can be assumed by either group at a given iteration. It follows that an agent in either group, say $a_i \in A$, can issue proposals towards the members of the opposite group, starting out from its first preference and moving to a subsequent one at each iteration. Moreover, members of both groups can exhibit both the behavior characteristic of men as well as that of women in the Gale-Shapley algorithm. In particular, if a proposal issued by agent $a_i \in A$ is accepted by its recipient $b_j \in B$, then a_i is engaged to b_j and has no reason to issue other proposals, as men do in Gale-Shapley's algorithm. In that case, we say that a_i is *content* with its current partner, as it has already exhausted all options to get engaged with a more highly preferred member of B. However, as both sides issue proposals, a_i may later receive a proposal from an agent $b_k \in B$, such that $pr_{a_i}[b_k] < pr_{a_i}[b_j]$, meaning that a_i prefers b_k to its current fiancé b_j. Then a_i will break its engagement to b_j and become engaged to b_k instead, as women do in Gale-Shapley's algorithm. Thereafter, b_j becomes single and resumes proposing to other members of A.

Later, b_k may receive a more tempting proposal from another member of A. Then b_k breaks up with a_i, hence a_i resumes proposing to members of B. The interesting question that arises then is at which position in a_i's preference list this resumption should be directed to. One might be tempted to think that the resumption can occur at $pr_{a_i}[b_j]$, the last rank to which a_i has already proposed. Yet if that were to happen, a_i would miss the opportunity to propose to potential spouses whom it may have rejected while it was engaged to b_k, i.e., spouses of rank $r \in [pr_{a_i}[b_k], pr_{a_i}[b_j]]$. Such opportunities should not be missed; thus, a_i should resume its proposals from rank $pr_{a_i}[b_k]$. Thus, when an agent resumes proposing after a forced breakup, the resumption should *not* commence from the point in its preference list it had previously reached, but from the rank of the eloping partner. Each agent a_i maintains two indices on its preference list:

- n_{a_i}, the **next target** to whom a_i will propose; and
- m_{a_i}, the **current fiancé** of a_i.

When a_i is *single*, then $m_{a_i} = \infty$. When a_i has to issue a proposal, it should propose to $\ell_{a_i}[n_{a_i}]$. If a_i breaks up with b_j in order to accept a proposal from b_k, then it sets $n_{a_i} = m_{a_i} + 1 = pr_{a_i}[b_k] + 1$. Later, after b_k breaks up with a_i, it sets $m_{a_i} = \infty$, since a_i is now single and proposes to $\ell_{a_i}[n_{a_i}]$, its next preference after b_k, and proceeds until it becomes engaged. On the other hand, a_i may receive a proposal from b_j, where $pr_{a_i}[b_j] > n_{a_i}$, while being single. In such circumstances, a_i accepts b_j's proposal, as a woman would do in the Gale-Shapley algorithm, even though there are still unexplored options more preferable than b_j in its preference list; a_i will then be *motivated* to keep proposing to those unexplored members of the opposite group.

From our discussion it follows that an agent can be in one of three states, shown in the table below: At the beginning, all agents are *single*, with $m = \infty$ and $n = 1$. When agent a receives a proposal from b, where $pr_a[b] = k > n_a$, and accepts, then $m_a = k$. As a has not yet reached position m_a of its preference list in terms of proposals, it is *motivated* to keep proposing to its unchecked preferences between n_a and m_a. After such proposals yield no positive response, n_a reaches m_a, hence a becomes *content* and ceases proposing.

Status	single	motivated	content
index	$m = \infty$	$m > n$	$m = n$

We now present the methods by which an agent evaluates and issues proposals (functions *EVALUATE* and *PROPOSE* respectively). Function $M(a_i)$ returns a_i's current partner, i.e., $\ell_{a_i}[m_{a_i}]$. For the sake of readability, we do not distinguish the case when a_i is single in this expression.

```
function EVALUATE($a$, $b$)                    ▷ $a$: acceptor, $b$: proposer
    $old = M(a)$
    if $a.m == \infty$ or $pr_a[b] < a.m$ then
        $old.m = \infty$                         ▷ Break up with old
        $a.m = pr_a[b]$                          ▷ Engage with new
        if $a.n > pr_a[b]$ then
            $a.n \leftarrow pr_a[b] + 1$
    else
        return false
    return true
```

```
function PROPOSE($a$)
    $old = M(a)$
    if $a.n < a.m$ then
        $b = \ell_a[a.n]$
        if EVALUATE($b, a$) then                 ▷ $b$ accepted $a$
            $old.m = \infty$                     ▷ Break up with old
            $a.m = a.n$                          ▷ Engage with new
        else
            $a.n = a.n + 1$                      ▷ $b$ rejected $a$
```

Algorithm 1 shows our methodology; its input is two groups, wherein each member agent a comes with its preference list and its inverse (ℓ_a and pr_a, respectively). At each iteration, one group is chosen to act as proposers, while the other acts as acceptors. Each proposer agent then issues its next proposal. The groups exchange roles across iterations for the sake of equity. We discuss more on this exchange later. The process goes on until all agents become content. In our approach, an agent's next target index n does *not* increase monotonically, as in Gale-Shapley's algorithm. In other words, the monotonicity that characterizes Gale-Shapley's algorithm and guarantees its termination to a stable solution is lost. Algorithm 1 does not provide an intrinsic termination guarantee. The following theorem shows that, *if* the algorithm terminates, then it terminates to a stable solution.

Algorithm 1 Equitable SMA

```
Input: $A, B$
Output: A stable matching $w$
$w = \emptyset$
$k = 0$
while not (everyone is content) do
    $k+ = 1$
    $P =$ PICK_PROPOSERS($A, B, k$)
    for all $p \in P$ do
        PROPOSE($p$)
for all $a \in A$ do
    $w = (a, M(a))$
return $w$
```

Theorem 1: If Algorithm 1 terminates, it finds a stable matching.

Proof: The proof is by reductio ad absurdum. Assume the algorithm is executed for groups A and B and terminates, yet the final matching M is unstable. Then M must contain at least two unstable couples (a_i, b_j), (a_k, b_ℓ) and a blocking pair (a_i, b_ℓ), where a_i prefers b_ℓ to b_j and b_ℓ prefers a_i to a_k, $pr_{a_i}[b_\ell] < pr_{a_i}[b_j]$ and $pr_{b_\ell}[a_i] < pr_{b_\ell}[a_k]$. Since both a_i and b_ℓ are eventually content, they must have issued at least two proposals to each other (once by a_i and once by b_ℓ), which were rejected. Without loss of generality, assume a_i rejected b_ℓ's proposal as it was engaged to a more preferred option, b_f. However, since a_i has ended up paired to a less preferred partner, b_f must have broken up with a_i. Then a_i must have proposed to, and been rejected by, b_ℓ. However, b_ℓ cannot have rejected that proposal and ended up content with a_k afterwards. A contradiction. ■

Example 1: Assume three men, m_1, m_2, and m_3, and three women, w_1, w_2, and w_2, have the following preference lists:

• m_1: w_2, w_1, w_3	w_1: m_1, m_2, m_3
• m_2: w_1, w_2, w_3	w_2: m_3, m_2, m_1
• m_3: w_1, w_3, w_2	w_3: m_2, m_1, m_3

Assume each side gets a chance to propose in every second iteration. In the first iteration, all men propose to their first preference. After women respond, we get pairs $(m_1, w_2), (m_2, w_1)$, while m_3 and w_3 remain single. In Step 2, each woman proposes to her first preference. While m_1 and m_2 reject the proposals from w_1 and w_3, respectively, m_3 accepts the proposal from w_2. Then the formed pairs are $(m_2, w_1), (m_3, w_2)$. In Step 3, men propose again; m_2 is content with his current fiancé, yet m_1 (single) and m_3 (not content with w_2) propose to their next preferences, w_1 and w_3, respectively; both accept, as w_1 prefers m_1 to m_2 and w_3 is single. Then the engagements become $(m_1, w_1), (m_3, w_3)$. In Step 4 it is the women's turn; w_1 is content, but w_2, being single, proposes to m_2, who accepts; w_3 also issues a proposal to m_1, who rejects her. The engagements now become $(m_1, w_1), (m_2, w_2), (m_3, w_3)$. Now all agents are content, hence the algorithm terminates. The achieved scores are $c(M) = 12$ and $d(M) = 0$. Gale-Shapley's algorithm, applied on the same problem instance, terminates at the stable solution $(m_1, w_2), (m_2, w_1), (m_3, w_3)$, with $c(M) = 12$ and $d(M) = 4$. Notably, our solution achieves better (lower) sex equality cost with the same egalitarian cost.

This simple yet powerful methodology achieves more equal results, allowing each agent to continue negotiations until a state is reached where each agent has achieved the best possible outcome for themselves. The agents keep proposing and evaluating offers from others, until they are no longer motivated to continue proposing, exactly as in the Gale-Shapley algorithm. However, the big difference in our case is that an agent may be acting as a proposer and as an acceptor in different algorithm steps, while in SMA each agent retain their role throughout the algorithm's execution. The idea that if agents of opposing groups behave similarly, more equal results will be achieved is commonly found in the literature (e.g. [14], [15]). However, as we will analyze in the following section, this approach may create a non-termination problem, since the repeatability of some preferences may lead to the endless repetitions of the algorithm's states. To this end, several approaches exploit the traditional algorithm to find a male-optimal stable solution and transform this matching in order to deliver more fair results (e.g. [12]). From our experience, the monotonic property retained by SMA, is essential both for the correctness and the termination of the algorithm. Any approach that tries to provide "fair" features to the problem while violating this monotonic property may either lead to not stable solutions and/or may not terminate. However, as we are discussing in the following section, we tackle these challenges by employing a simple aperiodic proposer picking function that breaks the repetitions in all tested problem instances.

B. The Problem of Circular Dependencies

We have established that, in case Algorithm 1 terminates, it achieves a stable solution too. As the algorithm offers both sides the opportunity to act as proposers, we expect such a stable solution to achieve good quality in terms of $c(M)$ and $d(M)$. However, it is not guaranteed that such a termination will eventually arise.

The algorithm terminates once all agents are content. Such contentment would be guaranteed to arise if all proposers' next target indexes were monotonically increasing, as in Gale-Shapley's algorithm. However, in Algorithm 1, an agent may return to the same target preference several times. If agents' preference lists follow a circular pattern, an infinite loop may arise, in which a group of agents continuously reissue the same proposals to, and break up with, each other. We illustrate an example of such circularity as follows.

Example 2: Assume two men, m_1, m_2, and two women w_1, w_2, are placed in each other's preference lists according to the following pattern:

- $m_1 : \ldots, w_1, \ldots, w_2$ $\quad$ $w_1 : \ldots, m_2, \ldots, m_1$
- $m_2 : \ldots, w_2, \ldots, w_1$ $\quad$ $w_2 : \ldots, m_1, \ldots, m_2$

Assume originally the only pair is (m_1, w_1). At a subsequent iteration, w_1 proposes to m_2, who accepts, hence we get (m_2, w_1). Later, m_2 proposes to w_2, who accepts. Hence we get (m_2, w_2). While now w_1 can return to m_1, she is still busy proposing to others. Later, m_1, still single, proposes to w_2, who accepts, deserting m_2. Hence we get (m_1, w_2). While m_2 can now return to w_1, she is busy making other proposals. Eventually, w_1 proposes to m_1, hence m_1 rejects w_2, and we get (m_1, w_1) again. Later m_2 proposes to w_1, who leaves m_1, hence we get (m_2, w_1) again. Later w_2 proposes to m_2; the whole process *may* continue in a cycle.

In the above, stable states *do* exist: both $(m_1, w_1), (m_2, w_2)$ and $(m_1, w_2), (m_2, w_1)$ are stable as far as these four agents are concerned. Moreover, we can obtain these stable states; for instance, if m_1 and m_2 propose to w_1 and w_2, respectively, at the same iteration, then we arrive at a stable state. The problem arises not due to the *pattern* of preferences per se, but due to the *timing* of agents proposing to each other across iterations. This timing depends on which side acts as proposers at each iteration. As long as the decision of who act as proposers is made in a state-dependent way, then, once the system arrives at a state in which it has previously been, it is guaranteed to enter an endless cycle. The same circumstances will lead to an endless repetition of the same chain of actions and back to the same state. It follows that, in order to avoid such cycles, we should devise a way to pick proposers in a manner defined by factors unrelated to the current problem state.

One way would be to pick proposers in a *randomized* fashion. Yet, in such a case, our experiments would not be strictly reproducible, casting doubts on the veracity of results. We then opt for picking proposers not as a function of the *problem state*, but as a function of the algorithm iteration. To guarantee fairness among the two sides, a *periodic* function would be fit for that purpose; however, the very periodicity of such a function is liable to lead to endless cycling in the state graph as well, in case the number of iterations within a cycle matches an integer number of periods. Therefore, we opt for a function that is: (i) deterministic but not state-dependent, (ii) leading to evenly distributed outcomes, and (iii) aperiodic. As we describe in the following section, the desired properties are provided by a composition of a trigonometric and a polynomial function.

C. Assigning Proposers

Our Equitable Stable Marriage Algorithm (ESMA) uses the PICK_PROPOSERS function shown below. Proposers are picked using the sign of the function $\sin(k^2)$, where k is the algorithm's iteration counter, leading to repeatable and evenly distributed, yet aperiodic sequences. If the sign of the function is positive, then the first group is picked; else the second is assigned as the proposer group.

```
function PICK_PROPOSERS($A$, $B$, $k$)
    if $\sin(k^2) \geq 0$ then
        return $A$
    else
        return $B$
```

The chosen function alternates due to its sin component, yet progressively decreases its oscillation interval as the iteration count k grows due to its quadratic component. The group chosen as proposers depends on the sign of this function.

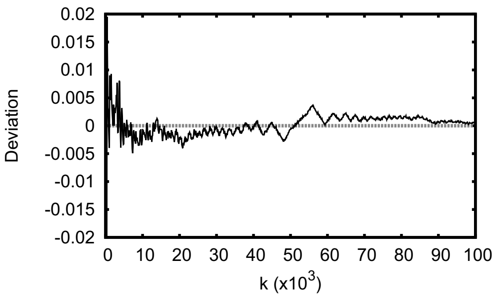

Fig. 1. Difference of appearances among proposing groups

It could be argued that, despite its oscillating nature, the function may not guarantee a "fair" treatment of the opposing groups. To investigate this matter, we measure how the *deviation*, defined as the difference between the times each side has been chosen as proposers, divided by the number of iterations k, evolves as a function of k. Figure 1 presents our findings. As we see, absolute deviation stays below 0.5% in the long run, while the favored side alternates as k grows. The choice of a polynomial and a trigonometric function as a proposer picking function leads all tested instances of the problem to termination. In our experimentation, we identified that the usage of a periodic function as a proposer picking function leads the algorithm to endless loops, since the period of the function may be synchronized with the repetition of the problem states.

D. Performance Optimization

As we have seen, an agent a reiterates over its preference list after a breakup using the next target index n. This provision guarantees the stability of the state the algorithm terminates at, yet it is also a computational burden. We discern that n does not need to be upgraded to the position of an eloping spouse. Assume n is the position of a's target index when it receives a proposal from $b = \ell_a[m]$, with $m < n$. Later, in case b elopes, a does not have to re-propose to all preferences between m and n. It suffices to propose to those suitors who had proposed to *a while* it was engaged with b, and whom it rejected as it was content; only those who have indicated interest need to be probed. To achieve this effect, it suffices to maintain a list of such suitors. This pruning of the proposal target list speeds up the overall algorithm.

IV. Experimental Evaluation

We now present our experimental study, which documents that ESMA *terminates* on the tested large problem instances and achieves good quality in terms of $c(M)$ and $d(M)$; in our experiments, these metrics are *normalized*, i.e., divided by n. The data sizes we employ greatly outnumber those used in experimental studies of related works [10]–[15], which were limited to $n \leq 200$. We use sizes of up to 2000, while we generate synthetic data of diverse skewness and type. Since many real-world problems studied in the literature (e.g. the Ministry of Education problem presented in [5]) entail a large number of agents, we chose a problem size that could be large enough to be realistic, yet leads Swing to terminate in a reasonable.

All algorithms were implemented in Java and ran on an Intel Xeon CPU at 2.00 GHz with 8GB RAM running Debian. We compare ESMA's performance in terms of the previously presented equality costs against the classical Gale-Shapley algorithm (SMA) and *Swing* [15]. Swing also allows both sides to propose. Yet it issues redundant proposals, recapitulating each agent's preferences anew at each iteration, and works in a simple alternating deterministic manner; thus, while ESMA requires only linear time per iteration, Swing requires quadratic time per iteration, and, moreover, easily falls into non-terminating cycles, as Everaere et al. recognize. We found that Swing does not terminate with 1% of $10,000$ randomly generated data sets of any size, while ESMA never encountered such a problem. We chose to compare ESMA against Swing because both algorithms guarantee that if termination is reached, the solution is stable. We emphasize that all our experiments are repeatable and our code is made available[1] so that any interested reader can try our algorithm on their own data.

For the experimental evaluation, we generate synthetic preference lists following one of the following distributions: (i) a *Uniform distribution*, creating preference lists by assigning scores uniformly at random, (ii) a *Gaussian distribution*, whereby preference lists are created starting from a *default* order where a_i has $score(a_i) = i$, adding Gaussian noise to those scores, and re-sorting and (iii) a distribution with *Discrete Regions*, in which the agents are partitioned in two disjoint sets of preferability: one set is annotated as the "hot region", indicating that the agents of this set are the most desirable in the opposite group and the "cold region" which is formed of the least desirable agents. In the Gaussian case, noise is multiplied by a skewness factor $S \leq n$. As S grows, the produced distribution becomes more like the uniform one. We use four datasets with S equal to 20%, 40%, 60% and 80% of n. Factor S is reversely proportional to the data *polarity*: the higher S is, the less polarized the data become. In the Discrete Regions case the agents of the same group are uniformly distributed in the preference lists of the opposing groups. We create four datasets in which the Hot Region consists of the 20%, 40%, 60% and 80% of n. We produced data of 20 different sizes, from 100 to 2000 agents per group, with 5 variants per size, while dropping data on which Swing could not terminate. The results we present for each data size are the averages of those runs.

A. Performance Evaluation

Figure 2 presents our results for the Uniform case. In terms of solution quality, Swing and ESMA perform similarly, as indicated both by Egalitarian and Sex Equality costs. SMA, on the other hand, achieves far worse quality in both metrics. Both Swing and ESMA perform up to 3 times better than SMA in terms of Egalitarian cost and up to 100 times better performance in terms of Sex Equality cost. However, ESMA outperforms Swing in terms of execution time; ESMA runs in approximately the same time as SMA, whereas Swing needs two orders of magnitude more time to complete. This result is due to two factors: (i) each iteration of Swing is much slower, as each agent reiterates proposals to all its preferences from the first to the current one (or until it finds a match), and (ii) the number of iterations Swing goes through is larger, even when it terminates, as the requirement that each agent aggressively reiterates proposals at each iteration leads to repetitive rounds of matchings. ESMA, on the other hand, does only one proposal per agent per iteration. To quantify this

[1] At https://github.com/equitable-stable-matching/esma/

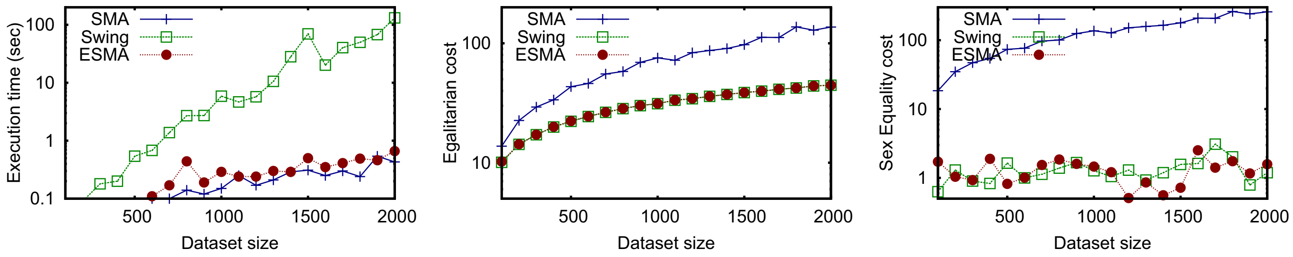

Fig. 2. Performance vs. data size, Uniform distribution

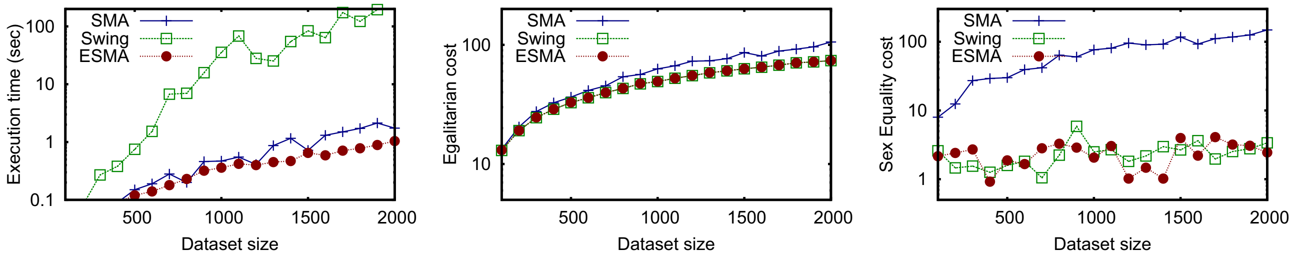

Fig. 3. Performance vs. data size, Gauss distribution

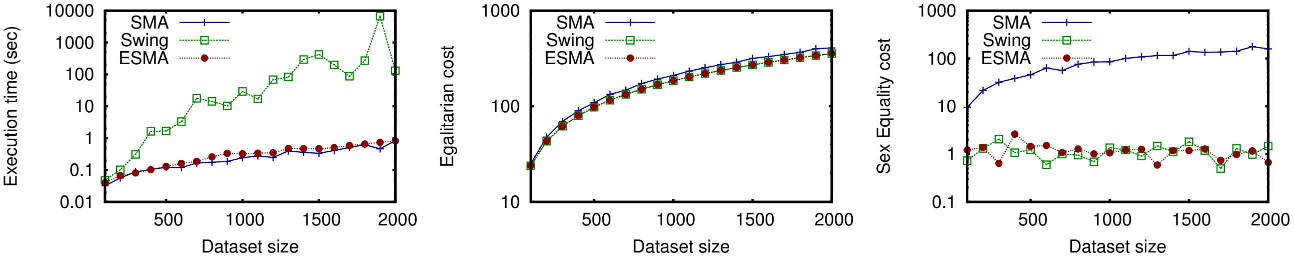

Fig. 4. Performance vs. data size, Discrete Regions distribution

difference, we measured that on 2000 agents, Swing needed approximately 23000 iterations, whereas SMA and ESMA completed in about 3000 steps. Figures are omitted due to space constraints. Interestingly, ESMA is faster than SMA in some cases (e.g. $n = 1900$). This can be attributed to the reason that, by allowing both sides to propose, a stable solution is reached in fewer iterations.

Figure 3 presents the performance of the three algorithms on the Gaussian dataset. The depicted experiments refer to the least polarized case (where $S = 80\%$). We again notice that the algorithms follow the same behavior, both in terms of equality and in terms of the execution time, as in the Uniform case (Figure 2). However, comparing Figures 2 and 3, we note that SMA lowers its difference from Swing and ESMA for both Egalitarian and Sex Equality cost. Furthermore, in Figure 4 we provide the respective results for the Discrete Regions dataset, in the case where the Hot Region covers 20% of n. We notice that the behavior of the three algorithms is equivalent to the Gaussian case. However, the difference of the equal algorithms and SMA in terms of Egalitarian cost is further decreased in this case, whereas in terms of Sex Equality cost both Swing and ESMA achieve "fair" results.

Taking the above observation further, we evaluate the performance of the algorithms on 2000 agents as the polarity of the data increases (S decreases). Figure 5 shows our findings. In terms of runtime, Swing is again the worst performer, while, surprisingly, ESMA gains an advantage over SMA; this is attributable to the fact that SMA may encounter difficulties with highly skewed preference lists, as it gives the initiative to one side only. Polarity affects the Egalitarian cost, with all algorithm's performance dropping as polarity grows. We attribute this effect to the fact that, due to the skewness of preference lists, some agents are universally more desirable, and less desirable ones tend to get matched to each other; thus global happiness worsens as skewness grows; on the other hand, sex equality improves for SMA, as polarity allows for more equity in a *one-side-optimal* matching, while it slightly worsens for others, as more polarized data lead to less equity compared to less polarized ones in a *balanced* matching.

V. Conclusion

This work revisited the NP-hard Equitable Stable Marriage Problem. We devised a novel solution, granting both sides the opportunity to propose in a minimalist and controlled

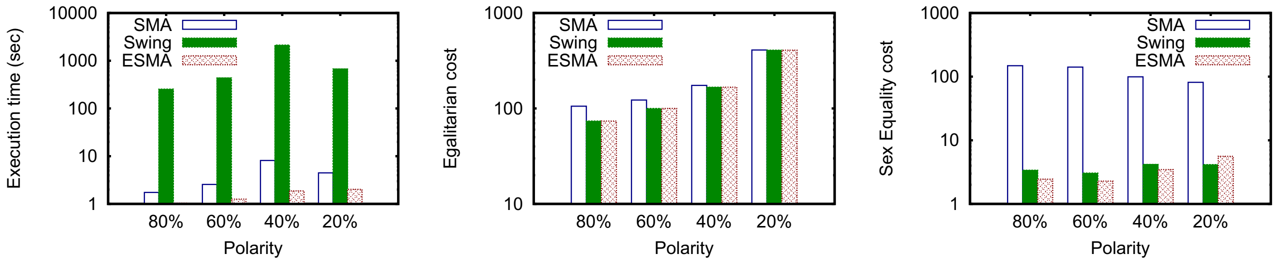

Fig. 5. Performance vs. polarity, Gauss distribution

manner, governed by an aperiodic, non-state-dependent function, so as to ensure repeatability in a machine-independent way. Our experimental study demonstrates that ours is, to our knowledge, the first practical algorithm that yields high-quality equitable stable matchings and terminates on all tested problem instances; it achieves performance similar to or better than that of Swing, the only existing practical algorithm for the problem, in terms of Sex Equality, whereas its execution time is similar to that of the classical Gale-Shapley algorithm that finds a solution biased in favor of one side.

VI. Acknowledgments

This work was partly supported by the European Commission in terms of the CELAR 317790 FP7 project (FP7-ICT-2011-8).

References

[1] D. Gusfield and R. W. Irving, *The Stable marriage problem - structure and algorithms*, ser. Foundations of computing series. MIT Press, 1989.

[2] D. E. Knuth, *Stable marriage and its relation to other combinatorial problems : an introduction to the mathematical analysis of algorithms*, ser. CRM proceedings & lecture notes. American Mathematical Society, 1997. [Online]. Available: http://opac.inria.fr/record=b1092346

[3] A. E. Roth, "The evolution of the labor market for medical interns and residents: A case study in game theory," *J. of Political Economy*, vol. 92, no. 6, pp. 991–1016, 1984. [Online]. Available: http://www.jstor.org/stable/1831989

[4] ——, "What have we learned from market design?" *The Economic Journal*, vol. 118, no. 527, pp. 285–310, 2008.

[5] C.-P. Teo, J. Sethuraman, and W.-P. Tan, "Gale-Shapley stable marriage problem revisited: Strategic issues and applications," *Management Science*, vol. 47, no. 9, pp. 1252–1267, 2001. [Online]. Available: http://dx.doi.org/10.1287/mnsc.47.9.1252.9784

[6] J. Liebowitz and J. Simien, "Computational efficiencies for multi-agents: a look at a multi-agent system for sailor assignment," *Electronic Government*, vol. 2, no. 4, pp. 384–402, 2005.

[7] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962. [Online]. Available: http://www.jstor.org/stable/2312726

[8] D. G. McVitie and L. B. Wilson, "The stable marriage problem," *Communications of the ACM*, vol. 14, no. 7, pp. 486–490, 1971.

[9] A. E. Roth and M. A. O. Sotomayor, *Two-sided matching: a study in game-theoretic modeling and analysis*, ser. Econometric Society monographs ; no. 18. Cambridge University Press, 1990.

[10] A. Kato, "Complexity of the sex-equal stable marriage problem," *Japan Journal of Industrial and Applied Mathematics*, vol. 10, no. 1, pp. 1–19, 1993. [Online]. Available: http://dx.doi.org/10.1007/BF03167200

[11] K. Iwama, S. Miyazaki, and H. Yanagisawa, "Approximation algorithms for the sex-equal stable marriage problem," *ACM Trans. on Algorithms*, vol. 7, no. 1, 2010.

[12] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Local search algorithms on the stable marriage problem: Experimental studies," in *ECAI*, 2010.

[13] M. Morge and G. Picard, "Privacy-preserving strategy for negotiating stable, equitable and optimal matchings," in *PAAMS*, 2011.

[14] P. Everaere, M. Morge, and G. Picard, "Casanova : un comportement d'agent respectant la privacité pour des mariages stables et équitables," *Revue d'Intelligence Artificielle*, vol. 26, no. 5, pp. 471–494, 2012.

[15] ——, "Minimal concession strategy for reaching fair, optimal and stable marriages," in *AAMAS*, 2013.

[16] R. W. Irving and P. Leather, "The complexity of counting stable marriages," *SIAM Journal on Computing*, vol. 15, no. 3, pp. 655–667, 1986.

[17] D. Gusfield, "Three fast algorithms for four problems in stable marriage," *SIAM Journal on Computing*, vol. 16, no. 1, pp. 111–128, 1987.

[18] R. W. Irving, P. Leather, and D. Gusfield, "An efficient algorithm for the "optimal" stable marriage," *Journal of the ACM*, vol. 34, no. 3, pp. 532–543, 1987.

[19] T. Feder, "A new fixed point approach for stable networks and stable marriages," *Journal of Computer and System Sciences*, vol. 45, no. 2, pp. 233–284, 1992.

[20] R. W. Irving, "Stable matching problems with exchange restrictions," *Journal of combinatorial optimization*, vol. 16, no. 4, pp. 344–360, 2008.

[21] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Procedural fairness in stable marriage problems," in *AAMAS*, 2011.

[22] B. Aldershof, O. M. Carducci, and D. C. Lorenc, "Refined inequalities for stable marriage," *Constraints*, vol. 4, no. 3, pp. 281–292, 1999.

[23] A. E. Roth and J. H. V. Vate, "Random paths to stability in two-sided matching," *Econometrica: Journal of the Econometric Society*, pp. 1475–1480, 1990.

[24] J. Ma, "On randomized matching mechanisms," *Economic Theory*, vol. 8, no. 2, pp. 377–381, 1996.

[25] É. Piette, M. Morge, and G. Picard, "Swing++: Méthode multi-agents pour la résolution du problème des mariages stables," in *Septièmes journées francophones Modèles Formels de l'Interaction (MFI'13)*, 2013.

[26] B. Zavidovique, N. Suvonvorn, and G. Seetharaman, "A novel representation and algorithms for (quasi) stable marriages." in *ICINCO*, 2005, pp. 63–70.

[27] K. Iwama, D. Manlove, S. Miyazaki, and Y. Morita, "Stable marriage with incomplete lists and ties," in *ICALP*, 1999.

[28] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Local search for stable marriage problems with ties and incomplete lists," in *PRICAI*, 2010.

[29] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Manipulation and gender neutrality in stable marriage procedures," in *AAMAS (1)*, 2009.

[30] ——, "Weights in stable marriage problems increase manipulation opportunities," in *TARK*, 2011.

[31] ——, "Manipulation complexity and gender neutrality in stable marriage procedures," *Autonomous Agents and Multi-Agent Systems*, vol. 22, no. 1, pp. 183–199, 2011.

[32] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Male optimality and uniqueness in stable marriage problems with partial orders," in *AAMAS*, 2010.

[33] H. Aziz, "Stable marriage and roommate problems with individual-based stability," in *AAMAS*, 2013.